



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,122	06/23/2003	Adam Wade Smith	13768.371	9110

47973 7590 04/23/2007  
WORKMAN NYDEGGER/MICROSOFT  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UT 84111

EXAMINER
----------

SHIN, KYUNG H

ART UNIT	PAPER NUMBER
----------	--------------

2143

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/23/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

## Office Action Summary

Application No.

10/602,122

Applicant(s)

SMITH ET AL.

Examiner

Kyung H. Shin

Art Unit

2143

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 23 June 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 23 June 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 12/21/04, 10/7/03.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

## DETAILED ACTION

1. This action is responding to application papers filed on **6-23-2003**.
2. Claims **1 - 30** are pending. Claims **1, 15, 28, 29** are independent.

### *Claim Rejections - 35 USC § 102*

3. The following is a quotation of 35 U.S.C. 102 which forms the basis for all anticipatory rejections set forth in this Office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims **1 - 30** are rejected under 35 U.S.C. 102(e) as being anticipated by **Singhal et al. (US Patent No. 7,096,418)**.

**Regarding Claim 1**, Singhal discloses in a server computer system that provides Web pages to requesting client computer systems, the Web pages potentially including content from locations that are external to the server computer system, the server computer system including a cache that stores portions of cached content previously received from locations external the server computer system, a method for causing a cache entry to be dependent on a customized dependency, the method comprising the following:

Art Unit: 2143

- a) an act of accessing an extensible cache dependency base class; (see Singhal col. 3, lines 28-33: cache utilization; col. 3, lines 42-44: col. 20, lines 53-60; col. 21, lines 33-37: software, program (i.e. method, class programming))
- b) an act of deriving a customized cache dependency class from the extensible cache dependency base class, the customized cache dependency class for implementing the customized dependency; (see Singhal col. 3, lines 28-33: implement cache dependency; col. 20, lines 53-60; col. 21, lines 33-37: software, program (i.e. method, class programming))
- c) an act of accessing a portion of content that is to be delivered to a client computer system; (see Singhal col. 3, lines 37-41: access content for client)
- d) an act of creating a cache entry that associates the customized dependency with the accessed portion of content; (see Singhal col. 3, lines 28-30: associate cache entry with dependent data) and
- e) an act of inserting the cache entry into cache such that the validity of the cache entry is dependent on the customized dependency. (see Singhal col. 3, lines 30-33: cache entry dependent on dependency data)

**Regarding Claim 2**, Singhal discloses the method set forth in claim 1 wherein the act of accessing an extensible cache dependency base class comprises an act of accessing a cache dependency base class that includes a notify dependency changed method such that classes derived from the cache dependency base class can implement purging functionality of the cache dependency base class by calling the notify dependency

Art Unit: 2143

changed method. (see Singhal col. 3, lines 33-37; col. 4, line 62 - col. 5, line 1:

invalidate (i.e. purge) cache content based on dependent data)

**Regarding Claim 3**, Singhal discloses the method set forth in claim 1, wherein the act of accessing an extensible cache dependency base class comprises an act of accessing a cache dependency base class that includes a dependency dispose method such that classes derived from the cache dependency base class can release resources occupied by a cache entry by calling the dependency dispose method. (see Singhal col. 8, lines 17-28: cache resources information updated (i.e. released))

**Regarding Claim 4**, Singhal discloses the method set forth in claim 1, wherein the act of deriving a customized cache dependency class from the extensible cache dependency base class comprises an act of deriving customized cache dependency class that can be implemented to cause a cache entry to be dependent on a database table. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 43-48: cache dependency, access content from database information)

**Regarding Claim 5**, Singhal discloses the method set forth in claim 1, wherein the act of deriving a customized cache dependency class from the extensible cache dependency base class comprises an act of deriving customized cache dependency class that can be implemented to cause a cache entry to be dependent on a Web service. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4,

Art Unit: 2143

lines 27-29: cache dependency, access information for web service (i.e. user request))

**Regarding Claim 6**, Singhal discloses the method as recited in claim 1, wherein the act of accessing a portion of content that is to be delivered to a client computer system comprises an act of accessing a portion of content that is to be included in a Web page. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 56-59: cache contents, dependency based on web page)

**Regarding Claim 7**, Singhal discloses the method as recited in claim 1, wherein the act of accessing a portion of content that is to be delivered to a client computer system comprises an act of accessing a portion of content from a database table. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 43-48: cache dependency, access content from database information)

**Regarding Claim 8**, Singhal discloses the method as recited in claim 1, wherein the act of accessing a portion of content that is to be delivered to a client computer system comprises an act of accessing a portion of content from a Web service. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 27-29: cache dependency, access information for web service (i.e. user request))

**Regarding Claim 9**, Singhal discloses the method as recited in claim 1, wherein the act of creating a cache entry that associates the customized dependency with the accessed

Art Unit: 2143

portion of content comprises an act of creating a cached that causes content from a database table to be dependent on the database table. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 43-48: cache dependency, create content based on database information)

**Regarding Claim 10**, Singhal discloses the method as recited in claim 1, wherein the act of creating a cache entry that associates the customized dependency with the accessed portion of content comprises an act of creating a cached that causes content from a database table to be dependent on an aggregate dependency. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 41-43: static content, dynamic content, multiple types of dependencies)

**Regarding Claim 11**, Singhal discloses the method as recited in claim 1, wherein the act of creating a cache entry that associates the customized dependency with the accessed portion of content comprises an act of creating a cached that causes content from a Web service to be dependent on the Web service. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 27-29: cache dependency based on web service (i.e. user request) information)

**Regarding Claim 12**, Singhal discloses the method as recited in claim 1, wherein the act of inserting the cache entry into cache such that the validity of the cache entry is dependent on the customized dependency comprises an act of inserting a cache entry

Art Unit: 2143

that is dependent on a database table. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 43-48: cache dependency based on database content information)

**Regarding Claim 13**, Singhal discloses the method as recited in claim 1, wherein the act of inserting the cache entry into cache such that the validity of the cache entry is dependent on the customized dependency comprises an act of inserting a cache entry that is dependent on an aggregate dependency. (see Singhal col. 3, lines 28-33: monitor cache dependency information; col. 4, lines 41-43: static content, dynamic content, multiple types of dependencies)

**Regarding Claim 14**, Singhal discloses the method set forth in claim 1, wherein a start time property does not need to be specified to invoke the functionality of the extensible cache dependency base class. (see Singhal col. 3, lines 28-33: cache dependency system; col. 21, lines 18-22: processor; col. 20, lines 53-60; col. 21, lines 33-37; col. 21, line 61 - col. 22, line 3: software, computer readable medium, program, (i.e. class, method software development))

**Regarding Claim 15**, Singhal discloses in a server computer system that provides Web pages to requesting client computer systems, the Web pages potentially including content from locations that are external to the server computer system, the server computer system including a cache that stores portions of cached content previously



Art Unit: 2143

received from locations external the server computer system, a method for purging a cache entry, the method comprising the following:

- a) an act of monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency; (see Singhal col. 3, lines 28-30: monitor dependency condition)
- b) an act of determining if the one or more custom dependency conditions have been be satisfied; (see Singhal col. 3, lines 30-33: determine if cache contents changed)
- c) an act of receiving an indication that the one or more custom dependency conditions have been satisfied; (see Singhal col. 3, lines 30-33: receive indication of dependency change) and
- d) an act of purging a cache entry at the server computer system in response to receiving the indication. (see Singhal col. 3, lines 33-37: invalidate cache contents reloaded)

**Regarding Claim 16**, Singhal discloses the method as recited in claim 15, wherein the act of monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency comprises an act of monitoring a database table dependency. (see Singhal col. 3, lines 28-33: monitor cache dependency; col. 4, lines 43-48: cache contents, dependency based on database content)

**Regarding Claim 17**, Singhal discloses the method as recited in claim 15, wherein the

Art Unit: 2143

act of monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency comprises an act of monitoring a Web services dependency. (see Singhal col. 2, lines 28-33: monitor cache dependency; col. 4, lines 27-29: web service)

**Regarding Claim 18**, Singhal discloses the method as recited in claim 15, wherein the act of monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency comprises an act of monitoring an aggregate dependency. (see Singhal col. 3, lines 28-30: monitor cache dependency information; col. 4, lines 41-43: cache dependency based on aggregated dependencies, static and dynamic cache content)

**Regarding Claim 19**, Singhal discloses the method as recited in claim 15, wherein an act of determining if the or more custom dependency conditions have been be satisfied comprises an act of determining in a flag entry has been incremented. (see Singhal col. 3, lines 30-33: change (i.e. incremented) in dependency value (i.e. flag value) or attribute)

**Regarding Claim 20**, Singhal discloses the method as recited in claim 15, wherein an act of determining if the or more custom dependency conditions have been be satisfied comprises an act of determining if a dependency condition associated with an aggregate dependency has been satisfied. (see Singhal col. 3, lines 30-37: dependency

Art Unit: 2143

condition processed; col. 4, lines 41-43: cache dependency based on aggregated dependencies, static and dynamic cache content)

**Regarding Claim 21**, Singhal discloses the method as recited in claim 15, wherein an act of determining if the one or more custom dependency conditions have been be satisfied comprises calling a notify dependency changed method upon the happening of some event specified by an instance of a customized cache dependency class. (see Singhal col. 3, lines 33-37: cache dependency, event processing completed)

**Regarding Claim 22**, Singhal discloses the method as recited in claim 15, wherein an act of receiving an indication that the one or more custom dependency conditions have been satisfied comprises an act of receiving an indication from a notify dependency changed method that the one or more custom dependency conditions have been satisfied. (see Singhal col. 3, lines 33-37: receive information concerning event condition to affect cache dependency)

**Regarding Claim 23**, Singhal discloses the method as recited in claim 15, wherein the act of purging a cache entry at the server computer system in response to receiving the indication comprises an act of purging a cache entry that was dependent on a database table. (see Singhal col. 3, lines 30-37: invalidate (i.e. purge) cache content due to change in dependency value; col. 4, lines 43-48: cache dependency, database contents)

**Regarding Claim 24**, Singhal discloses the method as recited in claim 15, wherein the act of purging a cache entry at the server computer system in response to receiving the indication comprises an act of purging a cache entry that was dependent on a Web service. (see Singhal col. 3, lines 30-37: invalidate (i.e. purge) cache content due to change in dependency value; col. 4, lines 27-29: web service (i.e. user request))

**Regarding Claim 25**, Singhal discloses the method as recited in claim 15, wherein the act of purging a cache entry at the server computer system in response to receiving the indication comprises an act of purging a cache entry that was dependent on an aggregate dependency. (see Singhal col. 3, lines 30-37: invalidate (i.e. purge) cache content due to change in dependency value; col. 4, lines 41-43: aggregate dependency, static and dynamic content)

**Regarding Claim 26**, Singhal discloses the method as recited in claim 15, further comprising: an act of releasing resources that were consumed to maintain the purged cached entry. (see Singhal col. 8, lines 17-28: resource release, update indices (i.e. parameters) designating cache dependency information)

**Regarding Claim 27**, Singhal discloses the method as recited in claim 24, wherein the act of releasing resources that were consumed to maintain the purged cached entry comprises an act of calling a DependencyDispose method associated with customized

Art Unit: 2143

cache dependency, the customized cache dependency being derived from an extensible cache dependency base class. (see Singhal col. 3, lines 33-37: invalidate (i.e. purge) cache contents based on cache dependency information; col. 21, lines 18-22: processor; col. 20, lines 53-60; col. 21, lines 33-37; col. 21, line 61 - col. 22, line 3: software, computer readable medium, program, (i.e. class, method software development))

**Regarding Claim 28**, Singhal discloses a computer program product for use in a server computer system that provides Web pages to requesting client computer systems, the Web pages potentially including content from locations that are external to the server computer system, the server computer system including a cache that stores portions of cached content previously received from locations external the server computer system, the computer program product for implanting a method for causing a cache entry to be dependent on a customized dependency, the computer program product comprising one or more computer-readable media having stored thereon computer executable instructions that, when executed by a processor, cause the server computer system to perform the following:

- a) access an extensible cache dependency base class; (see Singhal col. 3, lines 28-33: cache dependency capability)
- b) derive a customized cache dependency class from the extensible cache dependency base class, the customized cached dependency class for implementing the customized dependency; (see Singhal col. 3, lines 28-33:

Art Unit: 2143

cache dependency capability; col. 21, lines 18-22: processor; col. 20, lines 53-60; col. 21, lines 33-37; col. 21, line 61 - col. 22, line 3: software, computer readable medium, program)

- c) access a portion of content that is to be delivered to a client computer system; (see Singhal col. 3, lines 37-41: access content)
- d) create a cache entry that associates the customized dependency with the accessed portion of content; (see Singhal col. 3, lines 30-33: ) and
- e) insert the cache entry into cache such that the validity of the cache entry is dependent on the customized dependency. (see Singhal col. 3, lines 28-33: cache entry dependent on data attributes)

**Regarding Claim 29**, Singhal discloses a computer program product for use in a server computer system that provides Web pages to requesting client computer systems, the Web pages potentially including content from locations that are external to the server computer system, the server computer system including a cache that stores portions of cached content previously received from locations external the server computer system, the computer program product for implanting a method for purging a cache entry, the computer program product comprising one or more computer-readable media having stored thereon computer executable instructions that, when executed by a processor, cause the server computer system to perform the following:

- a) monitor one or more custom dependency conditions associated with an instance of a customized cache dependency; (see Singhal col. 3, lines 28-33: monitor

Art Unit: 2143

cache dependency information; col. 21, lines 18-22: processor; col. 20, lines 53-60; col. 21, lines 33-37; col. 21, line 61 - col. 22, line 3: software, computer readable medium, program)

- b) determine if the one or more custom dependency conditions have been be satisfied; (see Singhal col. 3, lines 30-33: determine cache dependency value)
- c) receive an indication that the one or more custom dependency conditions have been satisfied; (see Singhal col. 3, lines 33-37: determine cache dependency value) and
- d) purge a cache entry at the server computer system in response to receiving the indication. (see Singhal col. 3, lines 33-37: invalidate (i.e. purge) cache dependency entry)

**Regarding Claim 30**, Singhal discloses a Web page server comprising: a processor; and one or more computer readable media, having stored thereon: a cache; and a framework wherein the framework comprises: an extensible cache dependency base class for deriving custom dependency classes wherein the custom dependency classes are adapted to define at least one event the occurrence of which will cause a cache entry contained in the cached to be purged. (see Singhal col. 21, lines 18-22: processor; col. 20, lines 53-60; col. 21, lines 33-37; col. 21, line 61 - col. 22, line 3: software, computer readable medium, program; col. 3, lines 33-37: event processing, invalidate (i.e. purge) cache content due to dependency value information)

Art Unit: 2143

**Conclusion**

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kyung H. Shin whose telephone number is (571) 272-3920. The examiner can normally be reached on 9:30 am - 6 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David A. Wiley can be reached on (571) 272-3923. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.


Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

**K H S**

Kyung H Shin  
Patent Examiner  
Art Unit 2143

\*\*\*

KHS  
March 29, 2007

  
**DAVID WILEY**  
**SUPERVISORY PATENT EXAMINER**  
**TECHNOLOGY CENTER 2100**